

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A method comprising:

providing a common source code unit including production source code and test source code, the test source code having test methods for testing the production source code;

producing executable production code from the production source code and executable test code from the test source code; and

~~providing a global switch specifying whether to load the executable test code with the executable production code into a production environment~~

loading the executable test code with the executable production code in a development runtime environment in response to a setting of a global switch, wherein the development runtime environment executes the executable test code and performs the tests directed by the executable test code.

2. (Currently Amended) The method of claim 1 wherein the test source code and the production source code are written in a common programming language, and the test source code is identified by a key word that is recognized as a language extension to the common programming language, further comprising:

~~loading the executable production code in the production environment without the executable test code in response to a setting of the global switch.~~

3. (Currently Amended) The method of claim 1 wherein the executable test code includes test methods that can be called from test tools of the development runtime environment, further comprising:

~~executing the executable production code in a development environment; and~~

~~loading the executable test code with the executable production code in the development environment in response to a setting of the global switch.~~

4. (Currently Amended) The method of claim 1 wherein the executable test code includes test methods that test production methods of the executable production code by calling production methods of the executable production code, further comprising:

- ~~executing the executable production code in the production environment;~~
- ~~loading the executable production code in the production environment without the executable test code in response to a setting of the global switch;~~
- ~~changing the setting to specify that the executable test code is to be loaded with the executable production code in the production environment; and~~
- ~~loading the executable test code with the executable production code in the production environment, in response to the changed setting.~~

5. (Original) The method of claim 1 wherein the production source code and the test source code are generated using at least one of a procedural programming language including one of C, Fortran and Pascal, an object oriented programming language including at least one of a advanced business application program language (ABAP), Java programming language, C++ programming language and C# programming language.

6. (Currently Amended) The method of claim 1 further comprising checking that the production source code does not contain static references from the production source code to the test source code.

7. (Currently Amended) The method of claim 1 further comprising checking that the executable production code does not contain dynamic references from the executable production code to the executable test code.

8. (Currently Amended) The method of claim 1 wherein the executable production code and executable test source code is produced using a compiler that is operable to check the global switch to determine whether to generate an executable code unit having both executable production code and executable test code or an executable code unit without executable test code.

9-10. (Cancelled)

11. (Original) The method of claim 1 wherein the production source code and test source code are implemented in a unit test environment.

12. (Original) The method of claim 1 wherein the common source unit includes production source code and test source code sharing a same compilation unit.

13. (Original) The method of claim 1 wherein the common compilation unit includes executable production code and executable test code sharing a same compilation unit.

14. (Currently Amended) The method of claim 1 wherein the test ~~method includes methods~~ include a test assertion methods-method providing instructions for verifying an expected ~~state~~ operation of production source code.

15. (Currently Amended) A computer system comprising:

a common source unit having production source code and test source code with test methods for testing the production source code;~~and~~

a means for producing a common compilation unit having executable production code based on the production source code and executable test code based on the test class source code ~~or only the executable production code in response to a value of a system global switch; and~~

a means for loading the executable test code with the executable production code in a development runtime environment in response to a setting of a global switch, wherein the development runtime environment executes the executable test code and performs the tests directed by the executable test code.

16 . (Currently Amended) The system of claim 15 wherein the test source code and the production source code are written in a common programming language, and the test source code is identified by a key word that is recognized as a language extension to the common programming language, further comprising a means for:

- executing the executable production code in the production environment; and
- loading the executable production code in the production environment, without loading the executable test code, in response to the value of the global switch.

17. (Currently Amended) The system of claim 15 wherein the executable test code includes test methods that can be called from test tools of the development runtime environment, further comprising a means for:

- executing the executable production code in a development environment; and
- loading the executable test code with the executable production code in the development environment, in response to the value of the global switch.

18. (Currently Amended) The system of claim 15 wherein the executable test code includes test methods that test production methods of the executable production code by calling production methods of the executable production code, further comprising a means for:

- executing the executable production code in the production environment;
- loading the executable production code in the production environment, without loading the executable test code, in response to the value of the global switch;
- changing the value of the global switch to specify that the executable test code is to be loaded with the executable production code in the production environment; and
- loading the executable test code with the executable production code in the production environment, in response to the changed value of the global switch.

19. (Original) The system of claim 15 wherein the production source code and/or the test source code are generated using at least one of a procedural programming language including one of C, Fortran and Pascal, an object oriented programming language including at least one of a advanced business application program language (ABAP), Java programming language, and C++ programming language and C# programming language.

20. (Currently Amended) The system of claim 15 further comprising a means for checking that the production source code does not contain static references from the production source code to the test source code.

21. (Currently Amended) The system of claim 15 further comprising a means for checking that the executable production code does not contain dynamic references from the executable production code to the executable test code.

22. (Currently Amended) The system of claim 15 wherein the executable production code and/or executable test source code is produced using a compiler that is operable to check the global switch to determine whether to generate an executable code unit having both executable production code and executable test code or an executable code unit without executable test code.

23-24. (Cancelled)

25. (Original) The system of claim 15 wherein the production source code and test source code are implemented in a unit test environment.

26. (Original) The system of claim 15 wherein the common source unit includes production source code and test source code share a same compilation unit.

27. (Original) The system of claim 15 wherein the common compilation unit includes executable production code and executable test code sharing a same compilation unit.

28. (Currently Amended) The system of claim 15 wherein the test ~~method includes methods~~
include a test assertion methods-method providing instructions for verifying an expected ~~state~~
operation of production source code.

29-35. (Cancelled)

36. (New) A single source code unit comprising:

- production source code and test source code;
- the test source code having test methods for testing the production source code;
- the test source code and the production source code being written in a common programming language, wherein the test source code is identified by a key word that is recognized as a language extension to the common programming language;
- the test source code includes test methods that when compiled can be called from test tools of the development runtime environment;
- the test source code includes test methods that test production methods of the production source code by calling the production methods of the production source code;

37. (New) The single source code unit of claim 36, wherein the production source code does not contain static references from the production source code to the test source code.

38. (New) The single source code unit of claim 36, wherein the test source code includes a test method that calls a test assertion method to verify an expected operation of production source code.